**ANALYSIS REPORT GROUP-4**

**Executive Summary: -**

In this project, we are trying to classify whether one accident is fatal or not given enough data for some of the user columns. We used models SVM, Random Forest and Neural Networks for classifying the accident as fatal or nonfatal.

In conclusion, we can see that all models highly rely on the column ‘Injury’ to make majority of its classifications. All the three models are giving us good accuracies with SVM(96%), Random Forest(91%) and Neural Network (96%) and predicting the same outputs. This would further be improved by assigning weights to some of the important features. We can also use Hard Voting, an ensemble learning technique, to improve the model accuracy.

**Data Exploration: -**

We have 16860 Rows and 57 column in KSI.CSV

A screenshot of a computer

Description automatically generated with medium confidence

Below We can see the count, mean, std, percentile values, and max values for each column

Text

Description automatically generated

Correlation of each column with other has been showed below.

Greens blocks with very low correlation with other feature where as time and hour feature is highly related to each other.

A screenshot of a computer

Description automatically generated with medium confidence

ACCLASS which is the target class for the whole dataset are having 3 variations with fatal case, non fatal case and property damage only case. This plot shows the relation with these 3 classes.

Chart, bar chart

Description automatically generated

This bar plot also shows the relation of ACCCLASS variables with latitude occurrences.

Chart, bar chart

Description automatically generated

The number of various types of injury is shown here.

Chart, bar chart

Description automatically generated

Cases of no control traffic cases are most.

Chart

Description automatically generated with low confidence

Below are more analyses on the feature class columns in relation to ACCLASS which is the target class

We can see that most of the accidents occurred during Daylight irrespective of fatality. A picture containing chart

Description automatically generated

Chart, bar chart

Description automatically generated

Highest Number of Fatal accidents have occurred in Scarborough District and highest number of non-Fatal accidents have occurred in Toronto and East York.

Chart, waterfall chart

Description automatically generated

Many of the accidents resulting in fatality took place when road condition was other.

Chart, bar chart

Description automatically generated

Accidents result in fatality when the traffctl is no control mostly.

Text

Description automatically generated with medium confidence

**Feature Selection: -**

**Dropping** the columns which have numbers of different values can create complexity to process the data.

uniq\_cols=['ObjectId','ACCNUM','DATE','TIME','STREET1','STREET2','NEIGHBOURHOOD','WARDNUM','DIVISION']

Any “NULL”,” Unknown”, or” unknown” values are replaced with np. Nan so that we can figure out the missing values

Most of the missing values are from the column below so we have **dropped** them.

null\_cols=['OFFSET','FATAL\_NO','PEDTYPE','PEDACT','PEDCOND','CYCLISTYPE','CYCACT','CYCCOND','PEDESTRIAN', 'CYCLIST','MOTORCYCLE','TRUCK','TRSN\_CITY\_VEH','EMERG\_VEH','PASSENGER','SPEEDING','REDLIGHT',

'ALCOHOL','DISABILITY']

After dropping these columns, we have now 29 columns to be processed

**Data modelling: -**

**Processed columns**

Index(['X', 'Y', 'INDEX\_', 'YEAR', 'HOUR', 'ROAD\_CLASS', 'DISTRICT',

'LATITUDE', 'LONGITUDE', 'LOCCOORD', 'ACCLOC', 'TRAFFCTL', 'VISIBILITY',

'LIGHT', 'RDSFCOND', 'ACCLASS', 'IMPACTYPE', 'INVTYPE', 'INVAGE',

'INJURY', 'INITDIR', 'VEHTYPE', 'MANOEUVER', 'DRIVACT', 'DRIVCOND',

'AUTOMOBILE', 'AG\_DRIV', 'POLICE\_DIVISION', 'HOOD\_ID'],

dtype='object')

With these existing columns, we **impute** some of the missing values with the most frequent values.

We have separated the ACCLASS column from the feature columns.

ACCLASS is assigned as Y.

A screenshot of a computer

Description automatically generated with medium confidence

**Model Building:-**

**SVM Model(Developed by Chitra Hajra Roy):**

Data accuracy before fine-tuning the model of SVM is 0.86

**A screenshot of a computer

Description automatically generated with medium confidence**

After fine-tuning, the model of SVM is 0.96

**Text

Description automatically generated**

After fine-tuning the model of SVM with GridSearchCV is 0.96.

Parameters were

param\_grid= {'svc\_\_kernel': ['linear', 'rbf'],

'svc\_\_C': [0.01, 0.1, 1],

'svc\_\_gamma': [0.01, 0.06, 0.1]

}

The best estimator:

{'svc\_\_C': 1, 'svc\_\_gamma': 0.01, 'svc\_\_kernel': 'linear'}

Below the SVM model have prediction accuracy of the class (fatal or non fatal ) with KSI.CSV feature values.

**Graphical user interface

Description automatically generated**

Below we have mentioned the Featured values from KSI.CSV .With given inputs the prediction is 1 stands for fatal**.Graphical user interface

Description automatically generated**

**Random Forest(Developed by Kanishka Dhir):**

For Random Forest model, we are using RandomForestClassifier. Initially, we are using max\_depth as 5. This results in accuracy of 86%. We are fine tuning the model by using RandomizedSearchCV by giving number of trees between 10 to 200.Number of features to consider at every split is auto and sqrt.Maximum number of levels is in the range of 1 to 50.Minimum number of samples required to split a node is 2,5 or 10 and minimum number of samples required at each leaf node is 1,2 or 4. With the method of selecting samples from each tree is bootstrap = [True, False], we got an improved accuracy of 91%. This also gives us a very high precision of 99% and 38% recall and 55% score.

Initial Accuracy for Random Forest is 0.864

**Text

Description automatically generated**

Final Accuracy for Random Forest after Fine-tuning the model with RandomizedSearchCV is 0.9139976275207592

Confusion Matrix: -

[[ 261 431 0]

[ 2 4362 0]

[ 0 2 0]]

**Graphical user interface, text

Description automatically generated**

Best estimator for Random Forest is:-

Pipeline(steps=[('preprocessor',

ColumnTransformer(remainder='passthrough',

transformers=[('num',

Pipeline(steps=[('Scaler',

StandardScaler())]),

[]),

('cat',

OneHotEncoder(handle\_unknown='ignore'),

['X', 'Y', 'YEAR', 'HOUR',

'ROAD\_CLASS', 'DISTRICT',

'LATITUDE', 'LONGITUDE',

'LOCCOORD', 'ACCLOC',

'TRAFFCTL', 'VISIBILITY',

'LIGHT', 'RDSFCOND',

'IMPACTYPE', 'INVTYPE',

'INVAGE', 'INJURY',

'INITDIR', 'VEHTYPE',

'MANOEUVER', 'DRIVACT',

'DRIVCOND', 'AUTOMOBILE',

'AG\_DRIV', 'POLICE\_DIVISION',

'HOOD\_ID'])])),

('clf',

RandomForestClassifier(min\_samples\_leaf=2, min\_samples\_split=5,

n\_estimators=52, random\_state=57))])

**Graphical user interface, application

Description automatically generated**

Below we have mentioned the Featured values from KSI.csv. With given inputs the prediction is fatal.

**Graphical user interface

Description automatically generated**

**Neural Network (Developed by Vikas Trivedi):**

For the Neural network model, we are using the sklearn MLP classifier model with (10,20,5) hidden layers. This results in a model accuracy of 97%. This model has a high classification accuracy and precision. This is because the model weights for each perceptron in the hidden layer has been trained according to the features.

This model has also been hypertuned with different parameters and the best parameters for this model are determined as ‘tanh’ as the best solver and ‘adam’ as the best classifier.

Accuracy on Neural Network is 0.9472

**Text

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

Below we have mentioned the Featured values from KSI.csv. With given inputs the prediction is fatal.

**A picture containing graphical user interface

Description automatically generated**